MSIA 401 STATISTICAL METHODS FOR DATA MINING

Project Report

Fall Quarter 2014

BERK BOZOKLAR
DEMETRIOS FASSOIS
AMEER KHAN
AHSAN REHMAN

SECTION 1: EXECUTIVE SUMMARY

This report presents statistical analysis and modeling of data from an upscale online retail company. The company uses mailed catalogs to drive sales to its websites, and wishes to know what patterns of customer behavior determine whether customers respond to these catalogs, and if they do what is the amount of response.

The dataset was preprocessed to remove inconsistencies. The data model was developed in two parts – the first to predict which customers are most likely to make purchases in response to mailed catalogs, and the second to forecast the amount of money customers spend when they respond to the catalogs.

In order to identify potential respondents, the model uses customer attributes such as consistency of past purchases, variety of products ordered, and recency of the latest purchase. Forecast of the purchase amount of potential customers is based on their total spend on the website, the number of orders to-date and their spending patterns in the last two years.

From a pool of about 51,000 customers, the data model identifies approximately 5,000 potential respondents to the mailing campaign, 47% of whom are actual respondents. Analysis of the model shows that the website stands to earn a revenue of $112,000 if catalogs are mailed to the predicted potential customers alone. Thus, the model enables the website to lower its catalog costs by about 90%.

SECTION 2: INTRODUCTION

2.1 Overall Approach

The first step for data modeling was to understand the structure and layout of the data and the underlying assumptions. The data preprocessing included identifying and imputing missing values, as well as removing any logical inconsistencies from the dataset.

This was followed by exploratory analysis where we identified visual patterns in the data to examine the relationship between various the attributes. Histograms were used to identify the distribution of the dependent and independent variables, and determine which univariate transforms are suited to normalize their distributions. Scatterplots of the independent variables were used to understand the relationships between them to identify correlations. Plots of the dependent variable against the independent variables helped to select the most appropriate linearizing transformations for the predictors.

We also created new variables that captured additional customer information from the date variables, such as recency of last purchase and customer loyalty. Other variables were created to capture the average amount spent by the customer per order, as well as indicator variables to represent consistency and variety of purchases made by the customer.

The model building stage was divided into two segments: a classification model to predict which customers will respond to the catalog being mailed, and a linear regression model to estimate their expected purchase amount resulting from the catalog. Logistic regression was used for the classification. The model gives the probability of a customer responding to the catalog. Various factors such as purchase consistency, customer loyalty, recency of last purchase and overall sales history were included in the model. This model was challenging because the response variable is a rare event, since less than 10% of the customers responded positively to the catalog. This results in the model having very low sensitivity when the specificity is relatively high. This was overcome by selecting a low cutoff value that maximized specificity while not considerably lowering sensitivity.

The linear regression model was used to predict the purchase amount of each customer that responds positively to the catalog. This model used the order and sales history, as well as the recent sales pattern of the customer to predict their expected purchase amounts. The models were developed using variable selection methods in order to select the best predictors that fit the data. The models were evaluated using adjusted R^2 and the mean squared error of residuals. Model justification was

performed by checking for homoscedasticity, normality and multicollinearity. Once the models were finalized, outliers and influential points were removed to improve the fit.

The final step was to evaluate the performance of the models against the test set using the mean squared error of prediction (MSEP) and the Financial Criterion.

2.2 A Priori Hypotheses

The following a priori hypotheses were formulated before fitting the data:

- Recency of last purchase can be an important predictor of positive response to the catalog (classification model).
- Customer loyalty (the number of months between a customer was added and their last purchase) may correlate positively with their response to the catalog mailing.
- Consistency of orders and sales across the last few years may be important factors for purchase amount in response to the catalog.
- The customers are more likely to respond to the catalog if they have placed orders for both Fall and Spring.

2.3 Report Outline

The following sections cover the fitting and evaluation of models for predicting the purchase amounts of the customers. Section 3: Model Fitting covers Data Preprocessing, Variable Creation, Classification Models and Linear Regression Models. Section 4: Model Validation tests the models fit in Section 3 on the test set against the MSEP and the Financial Criterion. The results, key findings and recommendations are reported in the Conclusions section. The Appendix contains the R programming scripts, along with relevant plots and R console output.

SECTION 3: MODEL FITTING

3.1 Data Preprocessing

The data preprocessing stage included examining the data for missing values, understanding the structure and underlying assumptions in the data, and checking for logical inconsistencies and removing them to the extent possible. The following were the major steps performed at this stage:

a. Checking for Missing Values:

   The first step was to check for missing values within the data. Only the last purchase year variable, `lpuryear`, had `NA` values. These values were imputed by replacing them with the last digit of the year from the date of last purchase, `datelp6`, in order to maintain consistency. About 0.07% of the data was updated here.

b. Inconsistencies Within Date Variables (`datelp6` and `datead6`):

   The `datelp6` and `datead6` variables were converted to the date data type. The `lpuryear` was found to be inconsistent with the year from `datelp6` for about 7.5% of the data. Moreover, `lpuryear` stores only one digit of the last purchase year and is accurate within a year, while `datelp6` is accurate within 6 months. Hence, it was decided to drop `lpuryear` for developing models, and use the month and year values from `datelp6`.

c. Inconsistencies in Year-wise Order Variables with respect to Reference Year:

   The reference year for the year-wise order and sales variables is assumed to be July 2011 – June 2012.

   The year-wise order variables `ordtyr`, `ordlyr`, `ord2ago` and `ord3ago` were checked against `datelp6` for inconsistencies. For instance, `ordtyr` (orders this year) can be non-zero if `datelp6` is more recent than 2011-07-01 (the beginning of the reference year). This is because `datelp6` is accurate within 6 months. Thus, if `ordtyr` existed for customers with `datelp6` older than 2011-07-01, the `datelp6` was updated to the same. A similar process was followed for the remaining year-wise order variables. About 2% of the data points were affected by this correction.

   For all customers added in 2009 or later, it was checked if the order history and sales history variables, `ordhist` and `slshist`, added up to their year-wise variables. If the values did not match, `ordhist` was updated to be the sum of the year-wise variables.

d. Removing Blank Data Points:

All rows that contained blank entries in sales, order and the season (Fall and Spring) columns were removed from the data. 17 data points were completely blank and were removed.

e. Inconsistencies between Corresponding Order and Sales Variables:

The order variable for a specific year was set to zero if no sales existed for that year. This inconsistency is assumed to be due to returns. For instance, if `ordtyr` is non-zero and the corresponding `slstyr` is zero, we attribute this discrepancy to returns and set `ordtyr` to zero while also adjusting `ordhist` accordingly. The same process was repeated for other year-wise variables in order to remove the inconsistency between sales and orders. About 0.07% of the data was affected by this step.

The next step was to remove customers without any `ordhist` and `slshist` from the data since they do not contribute to any sales in the model. 269 data points were removed from the dataset at this step.

f. Inconsistencies within Season Variables:

The Fall and Spring orders were checked for consistency based on the assumption that all orders must be in either Fall or Spring categories. Thus, the `falord` and `sprord` variables must add up to `ordhist`. `ordhist` was updated when it was lesser than the sum of `falord` and `sprord`. This change affected about 3% of the data. This change was made assuming that `ordhist` was not updated when a sale occurred but one of the season order variables were.

3.2 Assessing the Nature of Relationships

a. Univariate Analysis

- Independent Variables

  Each variable's distribution was examined by plotting histograms in order to determine the kind of transformations needed to symmetrize the distribution, such as logarithmic, square-root or inverse transforms. All the numeric predictors displayed a right-skewed distribution, making logarithmic or square-root transforms a viable option. The most extreme values in the predictors were detected and removed from the dataset. About 17 data points were affected by this univariate outlier removal.

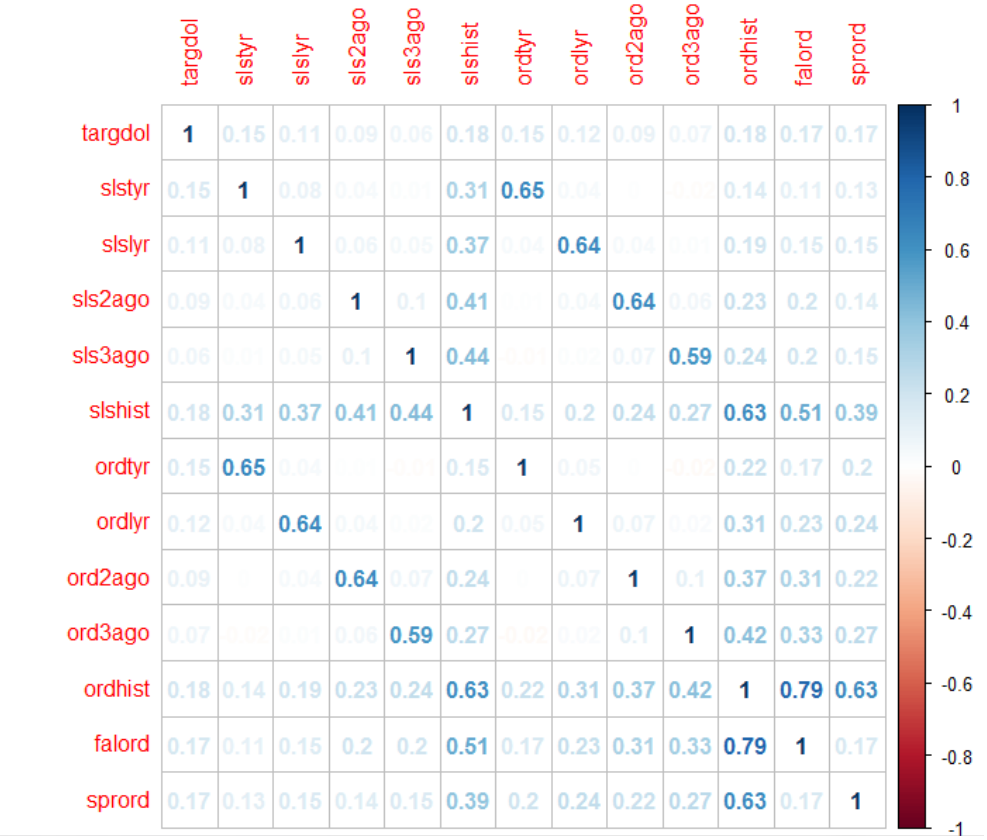
- Dependent Variable

  The dependent variable for the model is `targdol`. The `targdol` variable is sparse, with only 9% of the customers having non-zero values for the variable. Since `targdol` is an amount variable, it is right-skewed even when only its non-zero values are considered. Therefore, a logarithmic transform is most appropriate for this variable in order to normalize its distribution.

b. Bivariate Analysis

- Independent Variables

  Scatterplots of the independent variables were used to detect relationships between them. The correlation matrix was observed to see which variables have strong correlation amongst each other. By observing both the scatterplots and the correlation matrix results, strong correlation was discovered between the corresponding order and sales variables (for instance, `ordtyr` and `slstyr`). We also observe high correlation between `ordhist` and `falord`, which can potentially introduce multicollinearity in the model.



- Dependent Variable

  Scatterplots were made for dependent variable `targdol` against each predictor variable. This procedure is followed in order to see the predictor variables' relationship with the log transform of `targdol`, and what transforms can be used to linearize the relationship. The bivariate plots indicated logarithmic and square-root transforms to be a good fit for the sales variables and square-root and inverse transforms to work well for order variables.

3.3 Creating New Predictor Variables

In order to capture information from date variables about customers, new predictor variables were created. `recency_months` counts the number of months between the reference date 2012-09-01 and the date of last purchase, `datelp6`. `loyalty_months` is the number of months between the date of last purchase and the date the customer was added to the dataset. It is assumed that a customer can have a creation date later than the date of last purchase. This can apply when a customer makes a purchase and doesn't become a member until later and hence, this should not considered as an inconsistency. For this reason, `loyalty_months` is computed as the absolute value of the difference in the number of months between `datead6` and `datelp6`.

For the classification model, dummy variables were created for the order-related variables to indicate if a particular order variable was non-zero. For instance, if `ordhist` > 0 then `ordhist_bin` would be 1 since we are only interested in capturing the occurrence instead of the actual number. Interactions of the binary variables for year-wise orders indicate the consistency of customers. Corresponding dummy variables were also created for the season variables `falord` and `sprord`.

Another variable created was `slsPerOrd`, which measured the average sale per order for every customer that had non-zero `ordhist` in the dataset.

These variables were created in order to build more robust classification and linear models for the dataset. In order to avoid bloating the dataset, interaction variables for consistency of purchases, as well as for log and square-root transforms were dynamically created while fitting models.

3.4 Classification Models

Multiple logistic regression was used to build the classification model to predict whether a customer would respond to the catalog with a purchase. Since the dependent variable `targdol` is not binary, a dummy response variable `targdol_bin` is created that indicates if `targdol` is non-zero or not. The variables in this model are

selected on the basis of giving a good overall prediction of the customer responding to the catalog rather than the amount spent in response to the catalog.

The training dataset is separated from the test set and all models were built on the training set. The general approach for fitting a logistic model was to build a null model with no predictors, and use it in stepwise regression to select a model based on the AIC criterion. Two major approaches in terms of predictor transforms were to either use square-root transform for all the order and sales variables, or use dummy indicator variables for the order predictors, and logarithmic transforms for the sales ones.

3.4.1 Logistic Model 1

The model with square-root transforms is given below:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1\sqrt{recency} + \beta_2\sqrt{loyalty} + \beta_3\sqrt{ordtyr} + \beta_4\sqrt{ordlyr} + \beta_5\sqrt{ord2ago} + \beta_6\sqrt{falord} + \beta_7\sqrt{falord \times sprord} + \beta_8\sqrt{ordtyr \times ordlyr}$$

where π is the probability of success of `targdol_bin`.

The interaction term of $\sqrt{ordtyr \times ordlyr}$ represents consistency of the customer, while $\sqrt{falord \times sprord}$ shows versatility. The recency of purchases is incorporated using the term $\sqrt{recency}$. The summary of the model shows that all the predictors are highly significant:

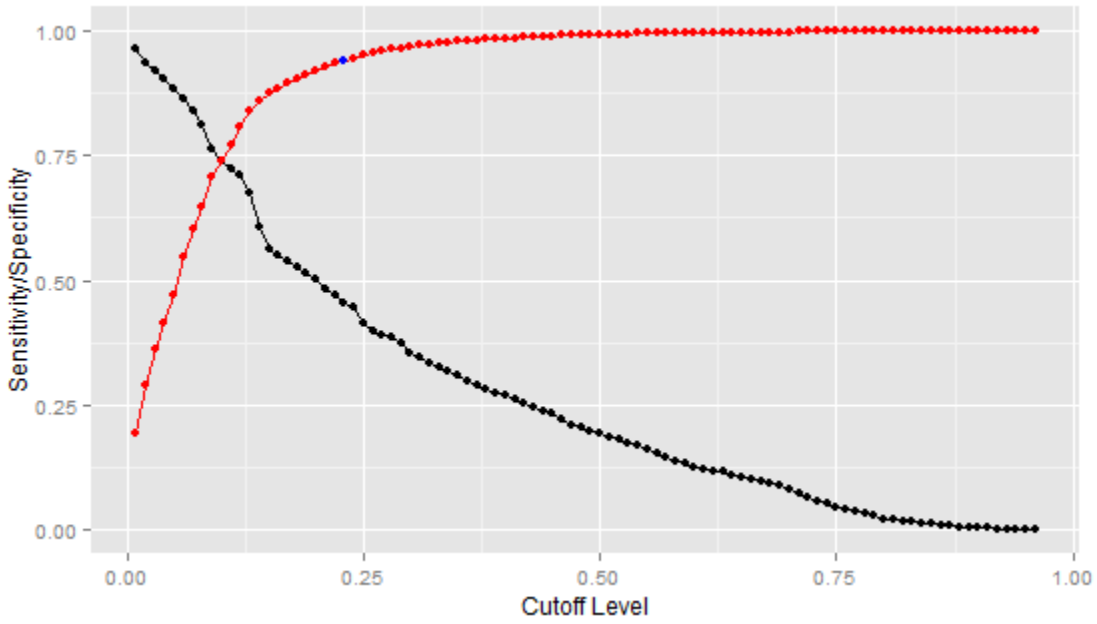
```
Call:
glm(formula = targdol_bin ~ sqrt(recency_months) + sqrt(loyalty_months
) + sqrt(falord):sqrt(sprord) + sqrt(falord) + sqrt(ordtyr) * sqrt(ord
lyr) + sqrt(ord2ago), family = "binomial", data = cat_logit)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                1.949279   0.126964  15.353  < 2e-16 ***
sqrt(recency_months)      -0.881934   0.020549 -42.919  < 2e-16 ***
sqrt(loyalty_months)       0.053622   0.004966  10.797  < 2e-16 ***
sqrt(falord)               0.600080   0.035928  16.702  < 2e-16 ***
sqrt(ordtyr)              -1.681979   0.073736 -22.811  < 2e-16 ***
sqrt(ordlyr)              -0.853130   0.052740 -16.176  < 2e-16 ***
sqrt(ord2ago)             -0.206045   0.036258  -5.683 1.33e-08 ***
sqrt(falord):sqrt(sprord)  0.073650   0.016673   4.417 9.99e-06 ***
sqrt(ordtyr):sqrt(ordlyr)  1.201791   0.065026  18.482  < 2e-16 ***

    Null deviance: 31823  on 50245  degrees of freedom
Residual deviance: 24935  on 50237  degrees of freedom
AIC: 24953
```

Model 1 Diagnostics: The p-value of the χ^2statistic is 0, indicating the model is significantly better than the null model. The VIF is computed to detect multicollinearity, and it is in the normal range for all the predictors.

sqrt(recency_ months)	sqrt(loyalty_ months)	sqrt (falord)	sqrt (ordtyr)	sqrt (ordlyr)	sqrt (ord2ago)	sqrt(falord): sqrt(sprord)	sqrt(ordtyr): sqrt(ordlyr)
3.74	1.84	2.34	5.66	2.98	1.36	1.88	3.36

From the ROC curve, the area under the curve obtained was 0.8168.

The most appropriate cutoff level for the model was selected by examining the plot of the sensitivity and specificity against the cutoff.



The above plot shows sensitivity (black) and specificity (red). As the cutoff increases, the sensitivity of the model steadily decreases, but there is a sharp increase in specificity. The cutoff value of 0.227 (blue dot) balances this tradeoff between lower sensitivity and higher specificity. The confusion matrix at this level is given by:

The confusion matrix for the above model for the fitted values at the 0.227 cutoff level is:

	Predicted 0	Predicted 1	Sum
Actual 0	42661	2756	45417
Actual 1	2601	2235	4836
Sum	45262	4991	50253

3.4.2 Logistic Model 2

The model obtained using binary order variables and logarithmic sales variables is given below:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1\sqrt{recency} + \beta_2\sqrt{loyalty} + \beta_3 ordtyr_{bin} + \beta_4 ordlyr_{bin} + \beta_5 ord2ago_{bin} + \beta_6 ord3ago_{bin} + \beta_7 falord_{bin} + \beta_8 \log(slshist + 1) + \beta_9(ordtyr_{bin} \times ordlyr_{bin})$$

where π is the probability of success of `targdol_bin`.

The interaction term $(ordtyr_{bin} \times ordlyr_{bin})$ indicates the consistency, while recency is given by $\sqrt{recency}$.

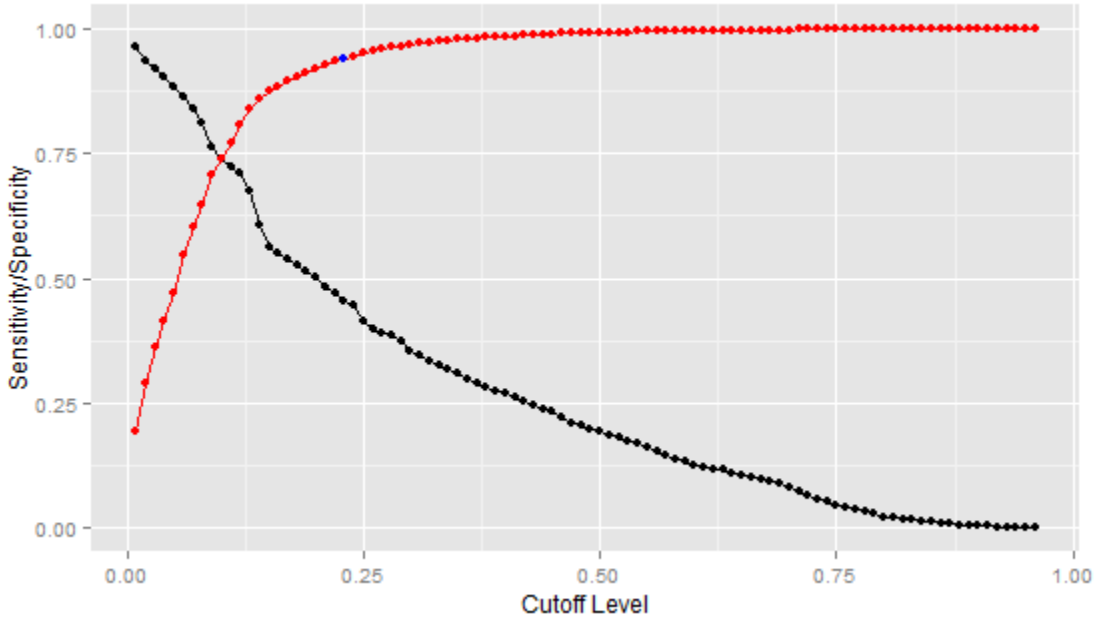
```
Call:
glm(formula = targdol_bin ~ sqrt(recency_months) + sqrt(loyalty_months
) + ordtyr_bin + falord_bin + ord3ago_bin + ord2ago_bin + log(slshist
+ 1) + ordlyr_bin + ordtyr_bin:ordlyr_bin, family = "binomial", data =
  cat_logit)

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.949193   0.148200  13.152  < 2e-16 ***
sqrt(recency_months)  -0.925608   0.021183 -43.695  < 2e-16 ***
sqrt(loyalty_months)   0.085703   0.004607  18.601  < 2e-16 ***
ordtyr_bin            -1.986917   0.082939 -23.956  < 2e-16 ***
falord_bin             0.613050   0.051301  11.950  < 2e-16 ***
ord3ago_bin            0.161239   0.040281   4.003 6.26e-05 ***
ord2ago_bin           -0.155496   0.039590  -3.928 8.58e-05 ***
log(slshist + 1)       0.067885   0.021611   3.141  0.00168 **
ordlyr_bin            -0.997987   0.060189 -16.581  < 2e-16 ***
ordtyr_bin:ordlyr_bin  1.737976   0.082002  21.194  < 2e-16 ***

    Null deviance: 31833  on 50252  degrees of freedom
Residual deviance: 24990  on 50243  degrees of freedom
AIC: 25010
```

Model 2 Diagnostics: The p-value of the χ^2 statistic is 0, so the model is significantly different from the null model. The VIF of the model is found to be in the normal range for all the predictors:

sqrt(recency _months)	sqrt(loyalty _months)	ordtyr _bin	falord _bin	ord3ago _bin	ord2ago _bin	log (slshist + 1)	ordlyr _bin	ordtyr_bin: ordlyr_bin
3.98	1.58	6.10	1.19	1.24	1.30	1.69	3.14	3.38

From the ROC curve, the area under the curve is 0.8187.

The most appropriate cutoff level for the model was selected by examining the plot of the sensitivity and specificity against the cutoff.



The above plot shows sensitivity (black) and specificity (red). As the cutoff increases, the sensitivity of the model steadily decreases, but there is a sharp increase in specificity. The cutoff value of 0.23 (blue dot) balances this tradeoff between lower sensitivity and higher specificity. The confusion matrix at this level is given by:

	Predicted 0	**Predicted 1**	**Sum**
Actual 0	42637	2780	45417
Actual 1	2632	2204	4836
Sum	45269	4984	50253

3.5 Multiple Regression Models

Linear regression models are used to predict the amount of money spent by customers (`targdol`) in response to the catalog. The training dataset for this modeling includes only the data points from the full training set that have `targdol` > 0.

Since `targdol` is a price variable, a logarithmic transform is used to normalize it. From the univariate and bivariate analysis on the predictor and response variables earlier, two sets of transforms for the order and sales variables are found to be appropriate: square-root and inverse transforms for the order variables, and square-root and logarithmic for the sales variables.

The general approach for fitting the models was to first select an initial model using a stepwise function over all possible predictors to determine `log(targdol + 1)`. The stepwise approach iteratively minimizes the AIC criterion to select the best model. The model resulting from this procedure was input to a best subsets selection function that also uses the stricter criterion, BIC.

3.5.1 Linear Model A

This model uses square-root transforms on all the sales and order predictors. The stepwise procedure returns the following model:

$$\log(targdol + 1) = \beta_0 + \beta_1\sqrt{ordhist} + \beta_2\sqrt{slshist} + \beta_3\sqrt{slstyr \times slslyr} + \beta_4\sqrt{slslyr}$$

All the β-coefficients of the model are highly significant at the 5% level. AIC of the model is 10691.85. This model was used as input to the best subsets regression procedure, and the following model was returned:

$$\log(targdol + 1) = \beta_0 + \beta_1\sqrt{ordhist} + \beta_2\sqrt{slshist} + \beta_3\sqrt{slstyr \times slslyr}$$

The `slslyr` variable was dropped by the best subsets, as it had the least contribution to reducing the error sum of squares (SSE). The summary of the final model is given below:

```
Call:
lm(formula = log(targdol + 1) ~ sqrt(slshist) + sqrt(ordhist) +
    sqrt(slstyr * slslyr), data = cat_lm)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.535295   0.027577 128.196  < 2e-16 ***
sqrt(slshist)         0.069353   0.002976  23.307  < 2e-16 ***
sqrt(ordhist)        -0.431683   0.022841 -18.899  < 2e-16 ***
sqrt(slstyr * slslyr) 0.002135   0.000438   4.873 1.13e-06 ***

Residual standard error: 0.7299 on 4830 degrees of freedom
Multiple R-squared:  0.1332, Adjusted R-squared:  0.1327
F-statistic: 247.4 on 3 and 4830 DF,  p-value: < 2.2e-16
```

The adjusted R^2 of the model is 13.27%. This model has a slightly higher AIC than the previous one given by the stepwise procedure (10694.97 vs. 10691.85), but has a lower BIC value (10727.39 vs. 10730.75).

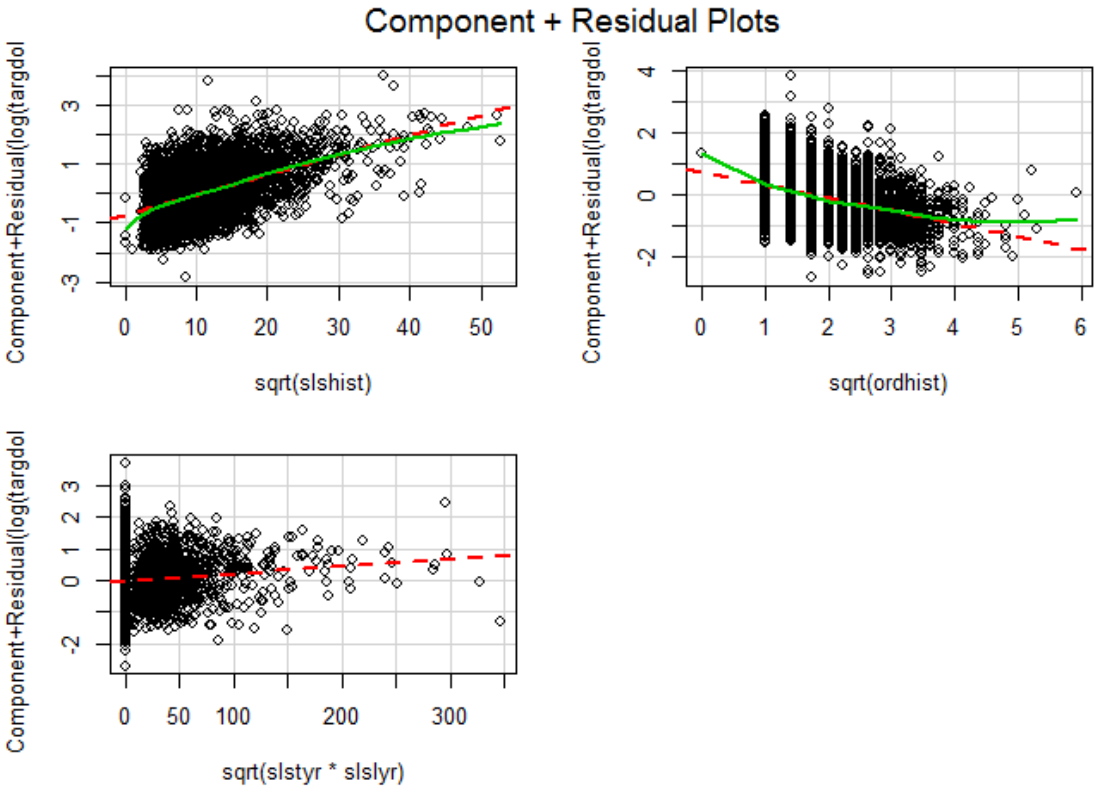
The VIF of the model is given below:

sqrt(slshist)	sqrt(ordhist)	sqrt(slstyr * slslyr)
3.98	1.58	6.10

Model A Diagnostics: The mean squared error of the model, MSE, is 0.534. The plot of the standardized residuals against the fitted values is randomly distributed and does not show any pattern, satisfying the homoscedasticity requirement. The Q-Q plot is approximately linear, indicating normality of the residuals.

The added-variable plot for the model is examined and we notice a linear relationship between the dependent variable and each predictor.

The component-plus-residual plot for the model is also examined, and we notice that the component and residual lines are close for each of the predictors. This shows that the square-root transform for the predictors is appropriate.



The data points with large residuals (outliers) and/or high leverage are removed using Cook's distance. Refitting the model improves the adjusted R^2 to 14.38%, and the MSE reduces from 0.534 to 0.46. We also notice a change in the VIF values:

sqrt(slshist)	sqrt(ordhist)	sqrt(slstyr * slslyr)
2.92	2.66	1.32

3.5.2 Linear Model B

This model uses inverse transforms on the order predictors, and logarithmic transforms on the sales variables. The stepwise procedure returns the following model:

$$\log(targdol + 1) = \beta_0 + \beta_1 \log(slshist + 1) + \beta_2 \frac{1}{ordhist + 1} + \beta_3 \log(slstyr \times slslyr + 1) + \beta_4 \log(slslyr + 1)$$

All the β-coefficients of the model are significant at the 5% level, except β_4. AIC of the model is 10725.31. This model was used as input to the best subsets regression procedure, and the following model was returned:

$$\log(targdol + 1) = \beta_0 + \beta_1 \log(slshist + 1) + \beta_2 \frac{1}{ordhist + 1} + \beta_3 \log(slstyr \times slslyr + 1)$$

The `slslyr` variable was dropped by the best subsets, as it had the least contribution to reducing the error sum of squares (SSE). The summary of the final model is given below:

```
Call:
lm(formula = log(targdol + 1) ~ I(1/(ordhist + 1)) + log(slshist +
    1) + log(slstyr * slslyr + 1), data = cat_lm)

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.022885   0.099734  10.256  < 2e-16 ***
I(1/(ordhist + 1))        2.298808   0.115148  19.964  < 2e-16 ***
log(slshist + 1)          0.401877   0.015794  25.444  < 2e-16 ***
log(slstyr * slslyr + 1) 0.016940   0.003883   4.362 1.32e-05 ***

Residual standard error: 0.7331 on 4832 degrees of freedom
Multiple R-squared:  0.1272, Adjusted R-squared:  0.1267
F-statistic: 234.8 on 3 and 4832 DF,  p-value: < 2.2e-16
```

The adjusted R^2 of the model is 12.67%. This model has a slightly higher AIC than the previous one given by the stepwise procedure (10726.76 vs. 10725.31), but has a lower BIC value (10759.18 vs. 10764.21).

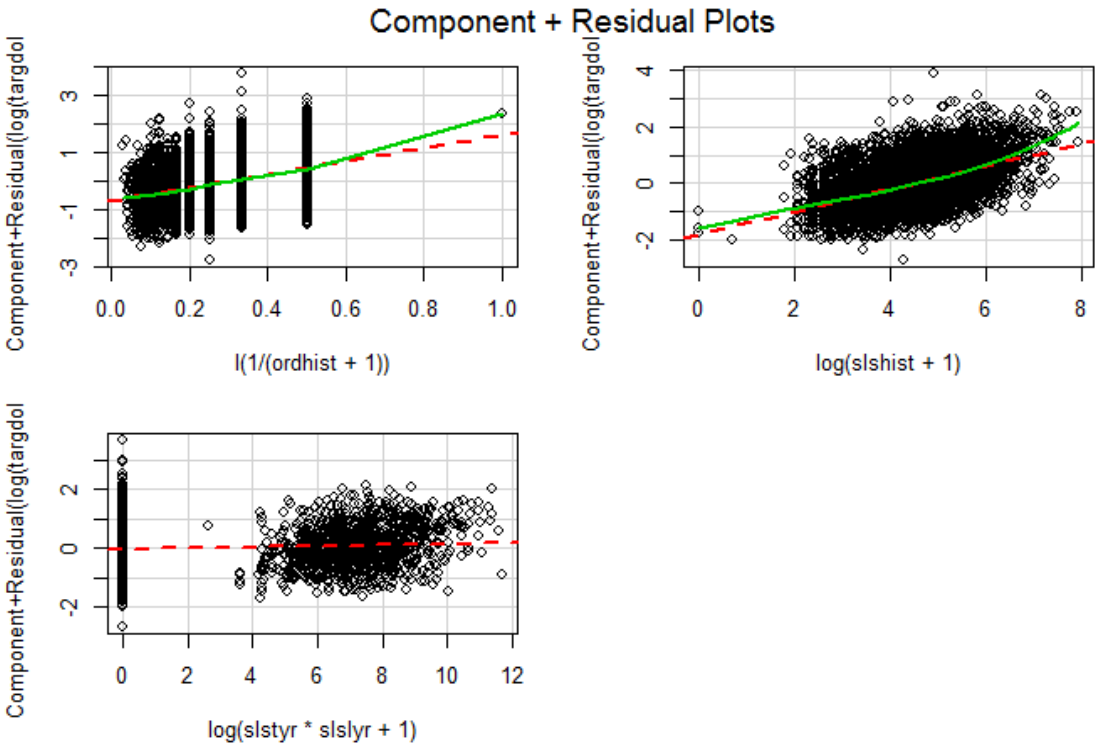
The VIF of the model is given below:

1/(ordhist + 1)	log(slshist + 1)	log(slstyr * slslyr + 1)
2.60	2.51	1.26

Model B Diagnostics: The MSE of the model is 0.54. The plot of the standardized residuals against the fitted values is randomly distributed and does not show any pattern, satisfying the homoscedasticity requirement. The Q-Q plot is approximately linear, indicating normality of the residuals.

The added-variable plot for the model is examined and we notice linear relationship between the dependent variable and each predictor.

The component-plus-residual plot for the model is also examined, and we notice that the component and residual lines are close for each of the predictors. This shows that the inverse transform for the order variables, and the log transform for the sales variables are apt.



The data points with large residuals (outliers) and/or high leverage are removed using Cook's distance. Refitting the model improves the adjusted R^2 to 15.41%, and the MSE reduces from 0.54 to 0.43.

SECTION 4: MODEL VALIDATION

4.1 Model Evaluation Metrics

In order to compute the validation of the model against the test set we computed the two evaluation measures: Mean Square Error of Prediction (MSEP) and the Financial Criterion.

- For calculating MSEP, we used the logistic regression model on the test dataset to identify potential data points with positive `targdol` values. The predicted `targdol` values were computed for these data points using the linear regression model. Because a log transform was used on `targdol` while fitting the linear model, the predicted values are reverse transformed as $e^{\hat{y}} - 1$ to obtain $\widehat{targdol}$. The MSEP is then computed using the formula

$$\frac{\sum_{i=1}^{n}\left(targdol_i - \widehat{targdol_i}\right)^2}{n-(p+1)}$$

  where $\widehat{targdol_i}$ is the predicted purchase amount and $targdol_i$ is the actual purchase amount of a customer i identified as a respondent to the catalog by the logistic model.

- The Financial Criterion was computed by first obtaining the predicted potential buyers using the classification model. These are the buyers to be targeted by mailing a catalog. The linear model then gives the predicted purchase amount of the targeted customers. Based on the predicted purchase amount, the top 5,000 customers are identified. The performance of the model is assessed by computing the actual purchase amount of these 5,000 customers.

4.2 Test Results for Logistic Regression Models

For Logistic Model 1 the confusion matrix for the predictions on the test set, using the previously specified cutoff level of 0.227 can be seen below:

	Predicted 0	**Predicted 1**	**Sum**
Actual 0	43500	2756	46256
Actual 1	2483	2237	4720
Sum	45983	4993	50976

The sensitivity achieved by this model is 47.39% and the specificity is 94.04%. The area under the curve for the test set is 0.8207.

Similarly for Logistic Model 2 the confusion matrix for the predictions on the test set is computed below:

	Predicted 0	Predicted 1	Sum
Actual 0	43439	2817	46256
Actual 1	2510	2210	4720
Sum	45949	5027	50976

The sensitivity achieved by this model is 46.82% and the specificity is 93.90%. The area under the curve for the test set is 0.8235.

4.3 Test Results for Linear Regression Models

The mean squared error of the predictions on the test set is computed using the formula

$$\frac{\sum_{i=1}^{n}\left(targdol_i - \widehat{targdol_i}\right)^2}{n-(p+1)}$$

where $\widehat{targdol_i}$ is the predicted outcome after performing the reverse transform $e^{\hat{y}} - 1$ and $targdol_i$ is the actual outcome of the test set for data point i.

The MSE for the test set is 2856.243 for Linear Model A.

The MSE for the test set is 2608.695 for Linear Model B.

4.4 MSEP and Financial Criterion

When Logistic Model 1 and Linear Model A are used to predict the target purchase amounts of customers, the MSEP is 2833.36. However, after the influential points are removed from Linear Model A using Cook's distance, the MSEP actually increased to 3706.05. This is in contrast to the fit becoming more robust on the training set when influential points were removed (as indicated by better MSE and adjusted R^2 values). This could be explained by the presence of many influential outlier values in the test dataset which result in large prediction errors, causing an increase in the MSEP.

The Financial Criterion when Logistic Model 1 is used to identify potential buyers and then the purchase amount of the potential buyers is estimated using Linear Model A is $112,219.40.

With Logistic Model 2 and Linear Model B, the MSEP is lower at 2118.45 but the Financial Criterion is also slightly lower at $108,843.00. After removing the influential points from Linear Model B, the MSEP decreases to 2079.38. This is because Linear Model B uses inverse transforms of the count variables, and log transforms of the sales predictors, which makes the model less susceptible to large outlier values in the test set. On the other hand, Linear Model A uses square-root transforms, which do not handle outliers in the test set as robustly as Linear Model B.

The results of the models are tabulated below:

	MSEP	Financial Criterion
Logistic Model 1 & Linear Model A	3706.05	$112,219.40
Logistic Model 2 & Linear Model B	2079.38	$108,843.00

CONCLUSION:

The team tested several models for the logistic and linear regression parts of the project. The final classification models selected account for customer consistency, recency and variety of purchases. Analyzing the β-coefficients of the logistic models, it is noticed that a customers with very recent purchases are more likely to respond to the catalog (`recency_months`), while customers who have been members of the website since a long time are less likely to respond (`loyalty_months`), going against our initial hypothesis. Consistent buying patterns over the last two years also increase the chances of customers buying (`ordtyr_bin` × `ordlyr_bin`). We also notice that customers who have Fall orders (`falord_bin`), and those with both Fall and Spring orders (`falord_bin` × `sprord_bin`) have higher odds of responding positively, confirming one of our a priori hypotheses.

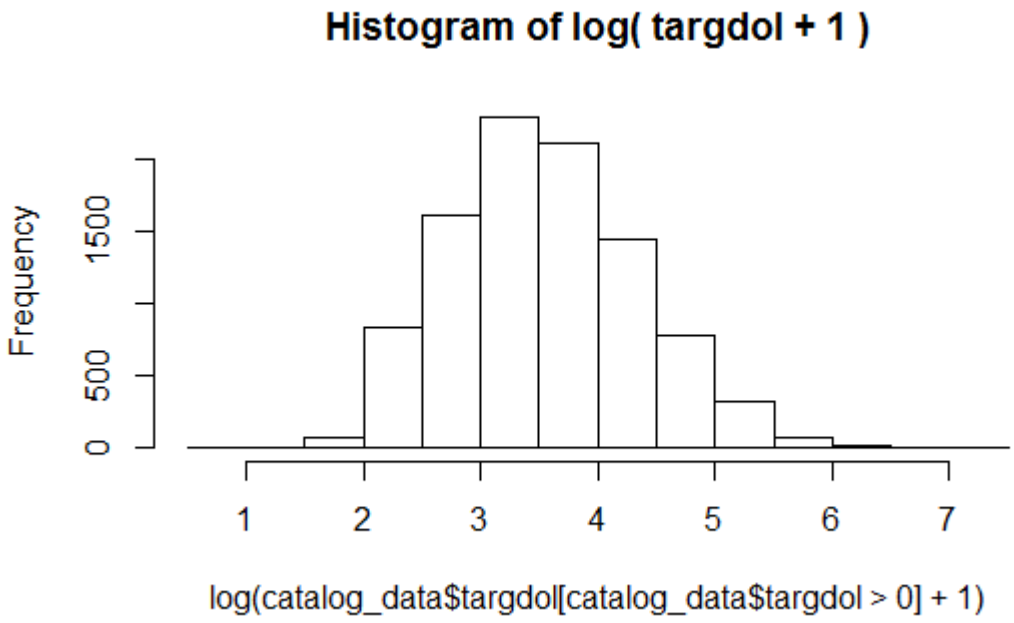
For the linear models, the most significant factors to determine the amount of money customers spend when they respond to the catalog were the total amount spent by customers in the past, the number of orders placed till date, and high spending patterns over the last two years. The β-coefficient for order history of a customer is negative (or positive for the inverse of `ordhist`), meaning customers with large number of orders will generally spend less amount of money. However, customers having a high total spend (`slshist`) are expected to spend more when they respond to the catalog. This is also true for those customers who spent a lot in the reference year and the year before (`slstyr` × `slslyr`).

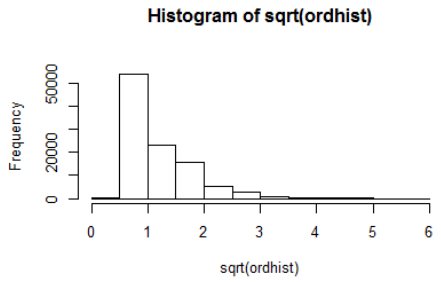
More accurate dates for customer creation and customer last purchases would have been key factors that would have helped in better predictions. Having past orders categorized by calendar years could also have provided better insights into a customer's order and spend patterns and hence, more accurate predictions for `targdol`. Other important predictors could have been whether the customers had previously responded to mailed catalogs or similar campaigns, frequency of their visits to the website and how they have rated their previous purchases.

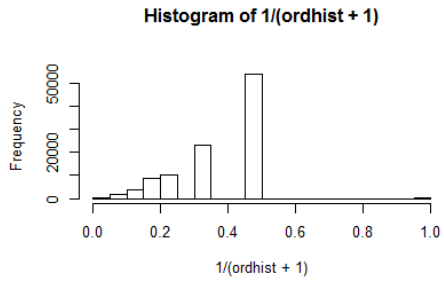
APPENDIX

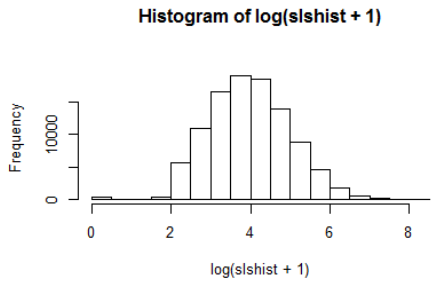
Plots for Model Analysis

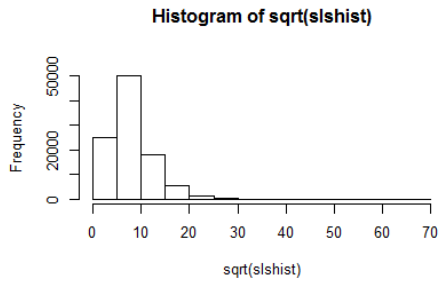
1. Plots for Univariate Analysis:

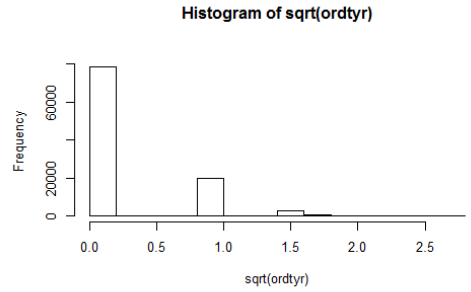


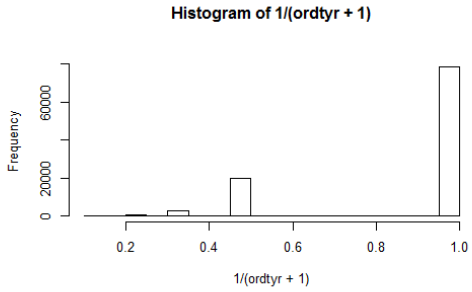


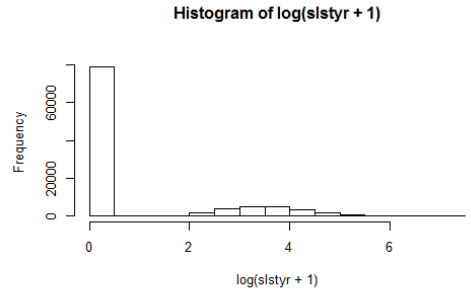


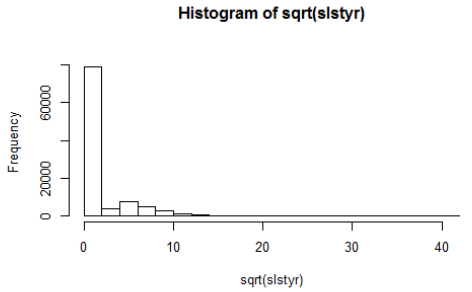




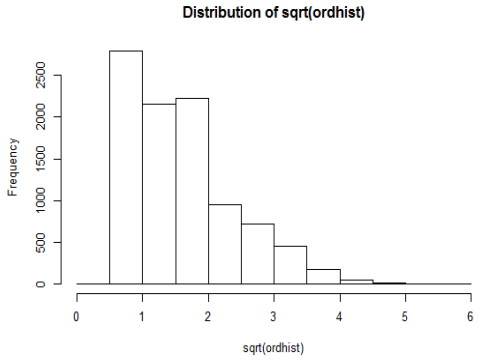


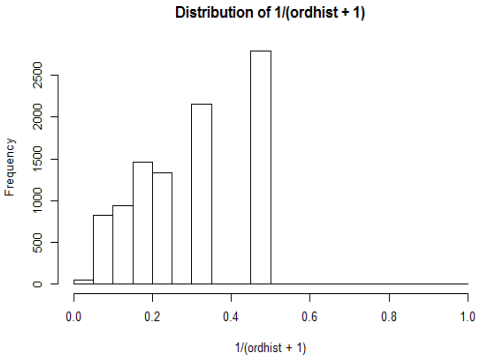


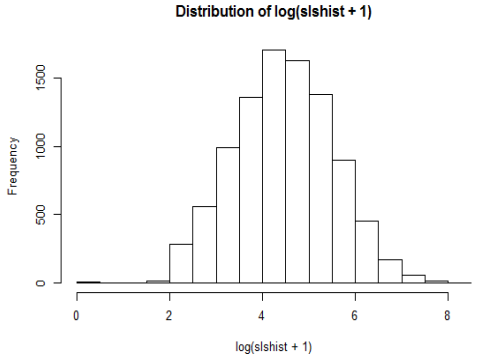


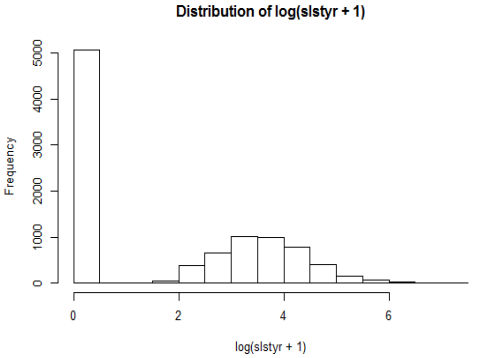


Histograms when Targdol > 0

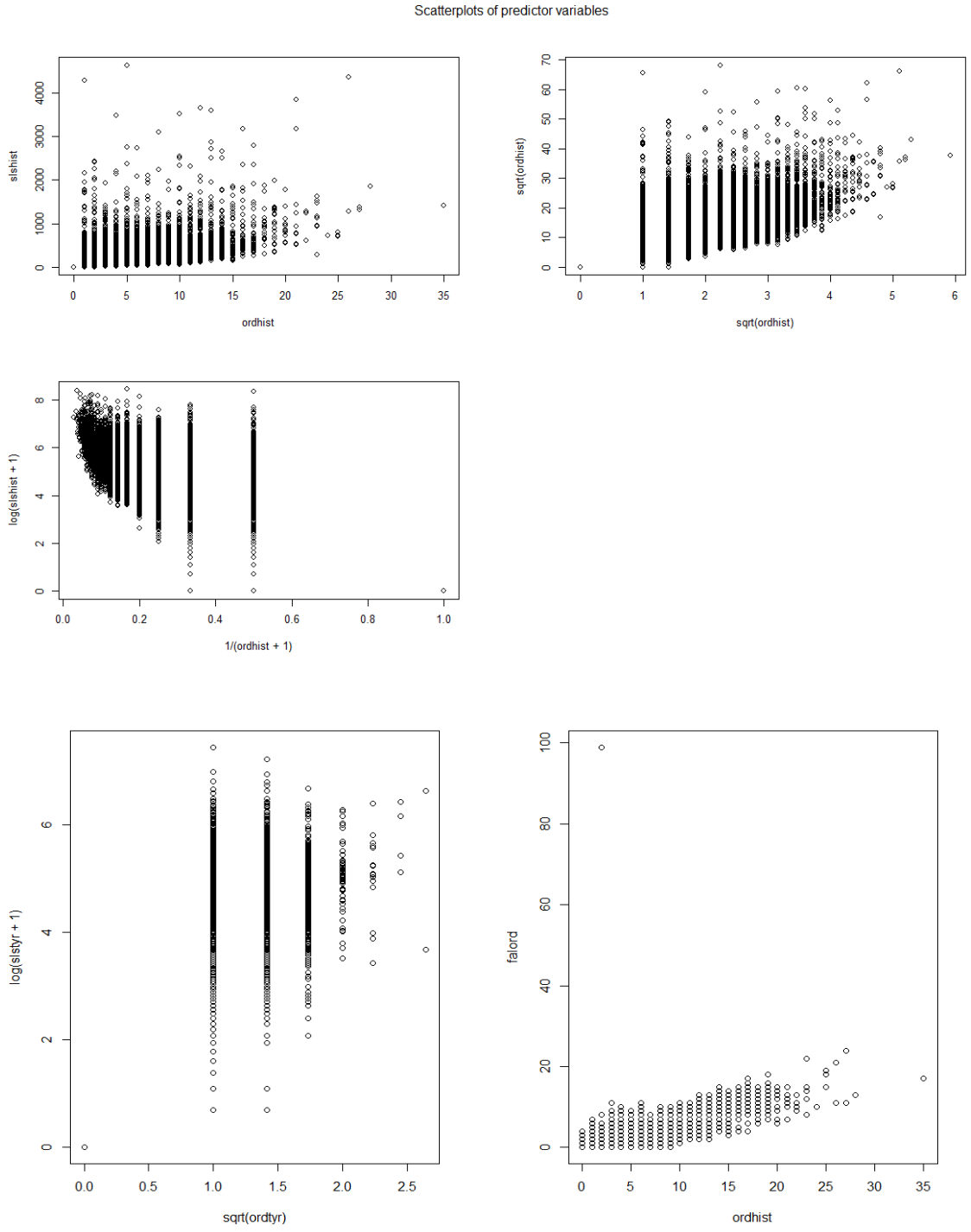




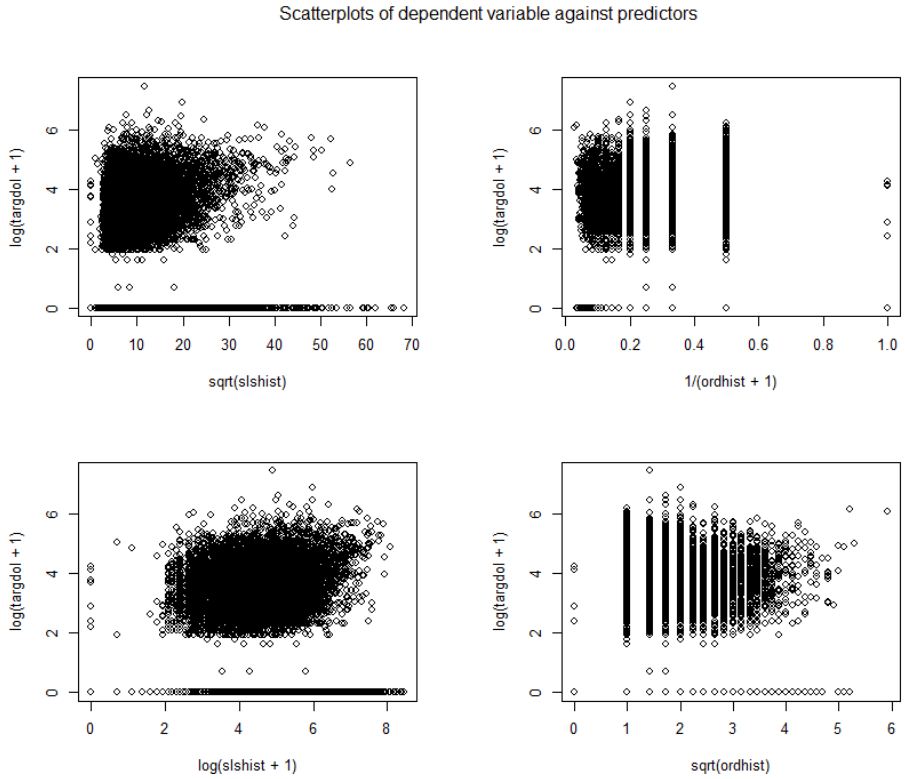




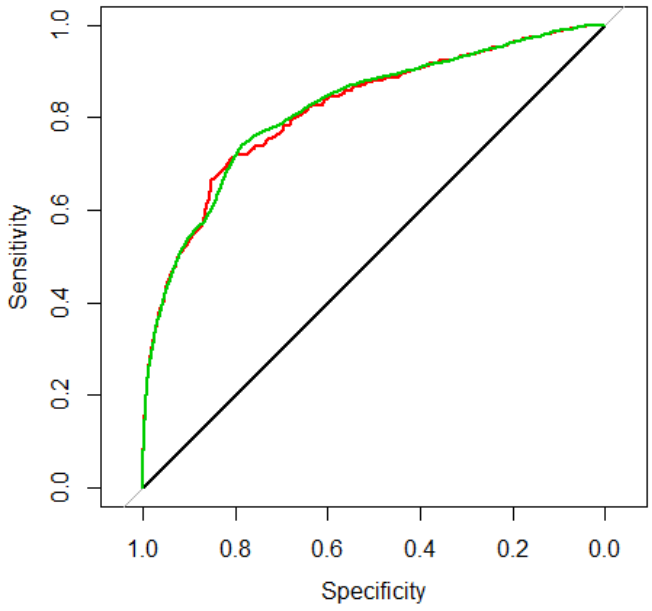
2. Scattterplots for Bivariate Analysis of Predictors:



3. Scatterplots of Dependent Variable against Predictor Variables:

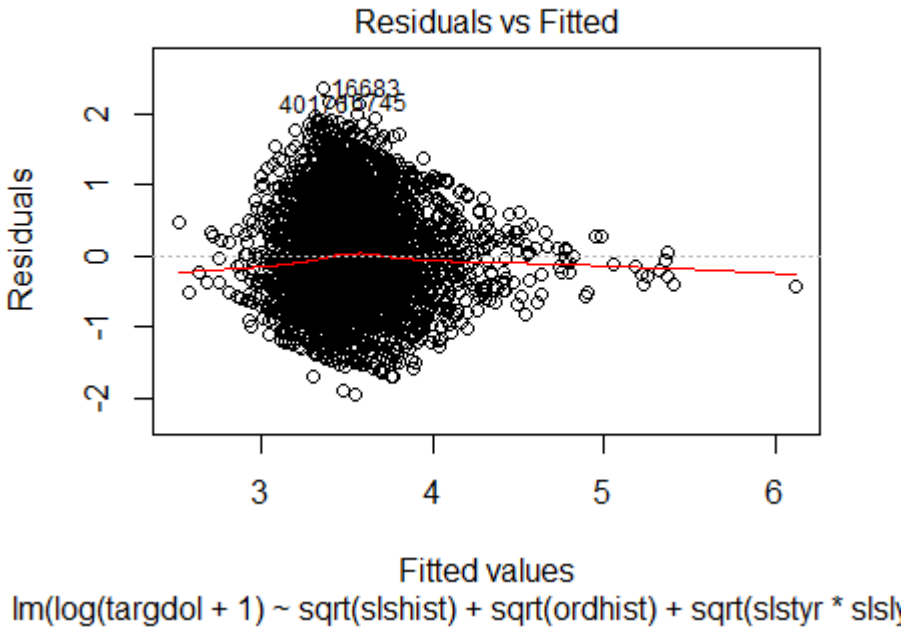


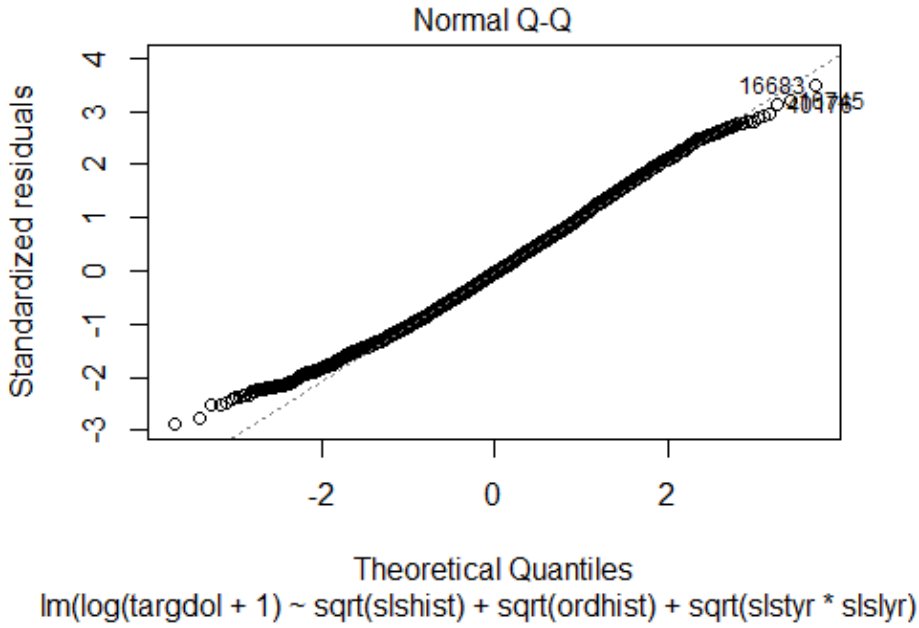
4. ROC Curves for Logistic Models 1 and 2:

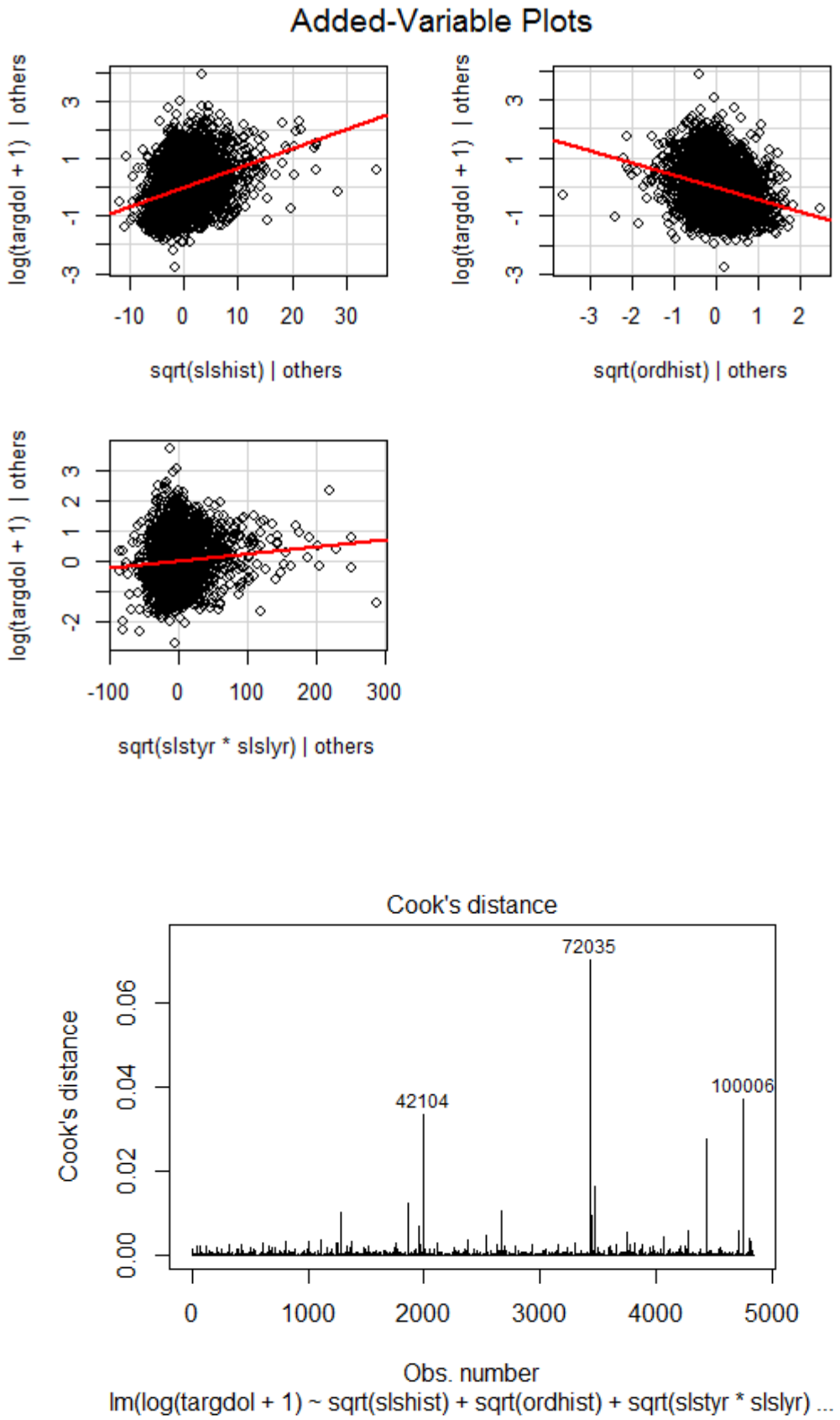


The red curve indicates the ROC for Logistic Model 1, and the green curve for Logistic Model 2.

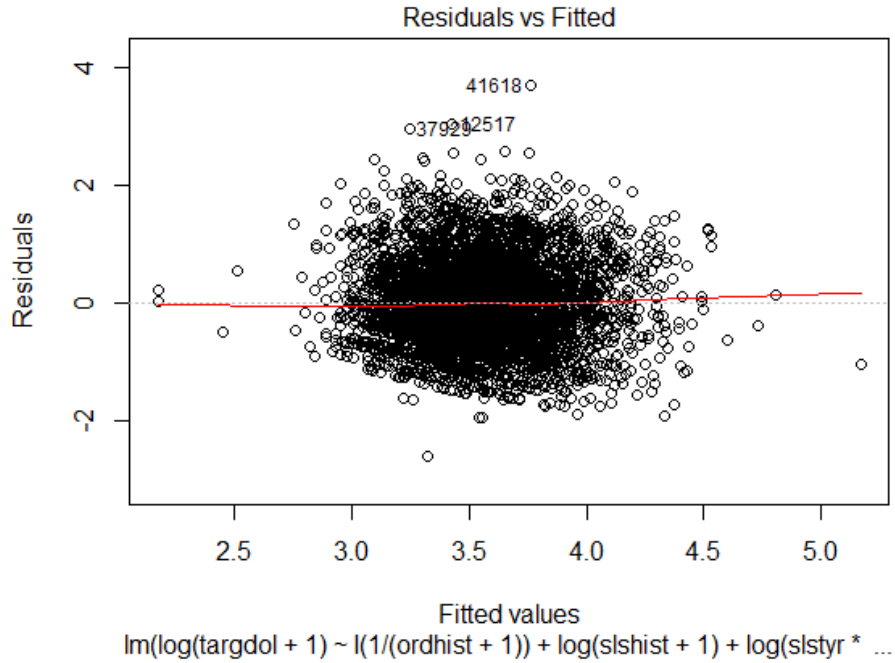
5. Plots for Linear Model A:

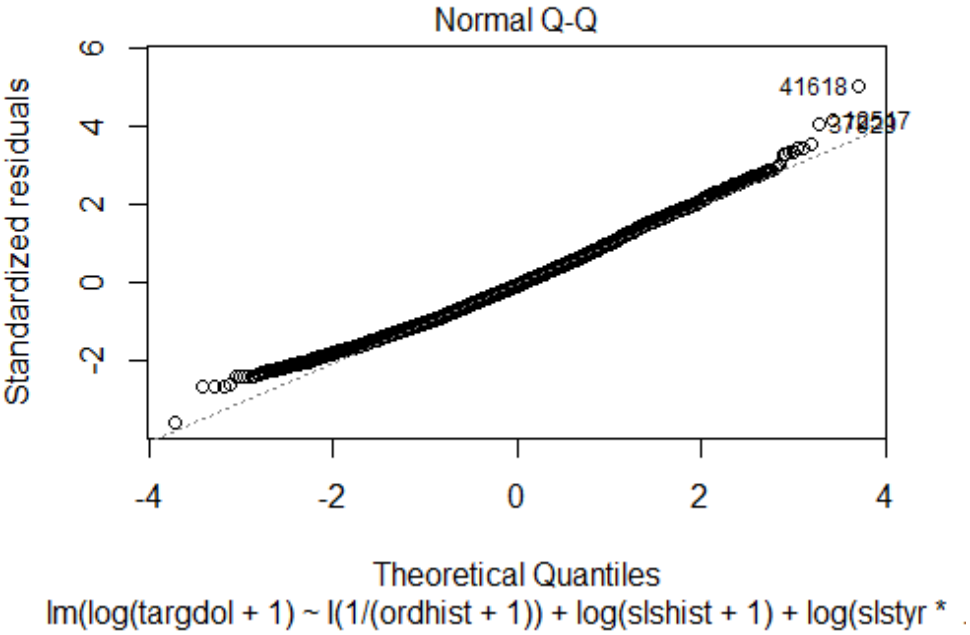


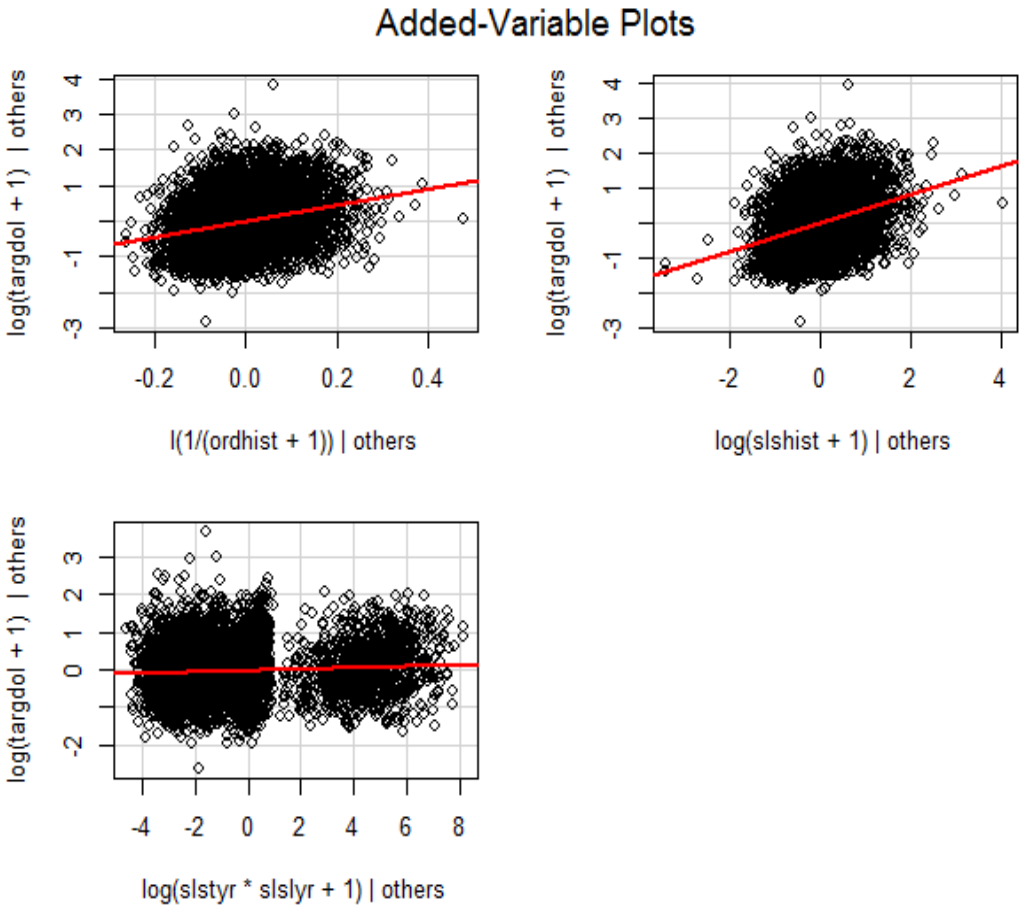



Added-Variable Plots
log(targdol + 1) | others
sqrt(slshist) | others
log(targdol + 1) | others
sqrt(ordhist) | others
log(targdol + 1) | others
sqrt(slstyr * slslyr) | others
Cook's distance
72035
42104
100006
Cook's distance
Obs. number
lm(log(targdol + 1) ~ sqrt(slshist) + sqrt(ordhist) + sqrt(slstyr * slslyr) ...

6. Plots for Linear Model B:






Added-Variable Plots
log(targdol + 1) | others
I(1/(ordhist + 1)) | others
log(slshist + 1) | others
log(slstyr * slslyr + 1) | others

R Scripts:

functions.r

```
# Functions to Compute Month Difference

months = function( date ) {
  date = as.POSIXlt( as.Date( date ) );
  return( 12 * date$year + date$mon + 1 );
}

months_diff = function( date1, date_ref ) {
  abs( months( date_ref ) - months( date1 ) );
}

# Function to Display Confusion Matrix for Logistic Regression

print_table = function( model, data = NULL, level = 0.5 ) {
  if ( is.null( data ) ) {
    tab = table( cat_logit[ , "targdol_bin" ], as.numeric( fitted(
model ) > level ) );
  }

  else {
    predicted = predict( model, newdata = data, type = "resp" );
    tab = table( data[ , "targdol_bin" ], as.numeric( predicted > level
) );
  }

  print( addmargins( tab ) );
  print( prop.table( tab, 1 ) );
  print( prop.table( tab, 2 ) );
}

# Function to compute MSEP

MSEP = function( fit_logit, fit_lm, data, level = 0.1 ) {
  predicted_logit = predict( fit_logit, newdata = data, type = "resp"
);
  data_logit = data[ predicted_logit > level, ];

  predicted_lm = predict( fit_lm, newdata = data_logit );
  predicted_lm = exp( predicted_lm ) - 1;

  msep = sum( ( data_logit[ , "targdol" ] - predicted_lm ) ^ 2 ) / (
nrow( data_logit ) - fit_lm$rank );
  print( msep );
}

# Function for Financial Criterion

fin_criterion = function( fit_logit, fit_lm, data, level = 0.1 ) {
  predicted_logit = predict( fit_logit, newdata = data, type = "resp"
);
  targeted_buyers = data[ predicted_logit > level, ];
```

```
  predicted_lm = predict( fit_lm, newdata = targeted_buyers );
  predicted_lm = exp( predicted_lm ) - 1;

  actual_buyers = targeted_buyers[ order( -predicted_lm ), ];
  actual_buyers = actual_buyers[ 1:5000, ];
  actual_response = sum( actual_buyers[ , "targdol" ], na.rm = T );
  print( actual_response );
}
```

data cleaning.r

```
library( car );
library( ggplot2 );
library( leaps );
library( corrplot );
library( pROC );

catalog_data = data.frame( read.csv( "catalog sales data for 2014
project.csv", header = T, stringsAsFactors = F ) );

# Convert to Date type
catalog_data = transform( catalog_data, datead6 = as.Date( datead6,
"%m/%d/%Y" ),
                          datelp6 = as.Date( datelp6, "%m/%d/%Y" ) );

#### Data Cleaning ####

# Check for NAs
sapply( catalog_data, function( x ) all( !is.na( x ) ) );

# Remove Blank rows
blank_index = apply( catalog_data[ , c( 1, 5:16 ) ], MARGIN = 1,
                     function( x ) all( x == 0 ) );

catalog_data = catalog_data[ !blank_index, ];

# Columns for years of date created and last purchase
catalog_data[ , "yearad" ] = as.numeric( format( catalog_data[ ,
"datead6" ], "%Y" ) );
catalog_data[ , "yearlp" ] = as.numeric( format( catalog_data[ ,
"datelp6" ], "%Y" ) );

# Replace NAs in lpuryear by last digit of datelp6
missing_index = is.na( catalog_data[ , "lpuryear" ] );
catalog_data[ missing_index, "lpuryear" ] = catalog_data[
missing_index, "yearlp" ] %% 10;

# Check for ordtyr and update yearlp
# Reference year is from Jul 2011 to Jun 2012 but datelp6 is accurate
within 6 months.
year_inconsistency = ( catalog_data[ , "ordtyr" ] & catalog_data[ ,
"datelp6" ] < "2011-07-01" );
catalog_data[ year_inconsistency, "datelp6" ] = "2011-07-01";
catalog_data[ year_inconsistency, "yearlp" ] = 2011;

year_inconsistency = ( catalog_data[ , "ordlyr" ] & catalog_data[ ,
"datelp6" ] < "2010-07-01" );
```

```
catalog_data[ year_inconsistency, "datelp6" ] = "2010-07-01";
catalog_data[ year_inconsistency, "yearlp" ] = 2010;

year_inconsistency = ( catalog_data[ , "ord2ago" ] & catalog_data[ ,
"datelp6" ] < "2009-07-01" );
catalog_data[ year_inconsistency, "datelp6" ] = "2009-07-01";
catalog_data[ year_inconsistency, "yearlp" ] = 2009;

year_inconsistency = ( catalog_data[ , "ord3ago" ] & catalog_data[ ,
"datelp6" ] < "2008-07-01" );
catalog_data[ year_inconsistency, "datelp6" ] = "2008-07-01";
catalog_data[ year_inconsistency, "yearlp" ] = 2008;

# Make orders for a specific year zero if no sales exist for that year
order_inconsistency = ( catalog_data[ , "ordtyr" ] & !catalog_data[ ,
"slstyr" ] );
catalog_data[ order_inconsistency, "ordhist" ] = catalog_data[
order_inconsistency, "ordhist" ] - catalog_data[ order_inconsistency,
"ordtyr" ];
catalog_data[ order_inconsistency, "ordtyr" ] = 0;

order_inconsistency = ( catalog_data[ , "ordlyr" ] & !catalog_data[ ,
"slslyr" ] );
catalog_data[ order_inconsistency, "ordhist" ] = catalog_data[
order_inconsistency, "ordhist" ] - catalog_data[ order_inconsistency,
"ordlyr" ];
catalog_data[ order_inconsistency, "ordlyr" ] = 0;

order_inconsistency = ( catalog_data[ , "ord2ago" ] & !catalog_data[ ,
"sls2ago" ] );
catalog_data[ order_inconsistency, "ordhist" ] = catalog_data[
order_inconsistency, "ordhist" ] - catalog_data[ order_inconsistency,
"ord2ago" ];
catalog_data[ order_inconsistency, "ord2ago" ] = 0;

order_inconsistency = ( catalog_data[ , "ord3ago" ] & !catalog_data[ ,
"sls3ago" ] );
catalog_data[ order_inconsistency, "ordhist" ] = catalog_data[
order_inconsistency, "ordhist" ] - catalog_data[ order_inconsistency,
"ord3ago" ];
catalog_data[ order_inconsistency, "ord3ago" ] = 0;

# Remove customers without order or sales history
catalog_data = catalog_data[ !( catalog_data[ , "ordhist" ] == 0 &
catalog_data[ , "slshist" ] == 0 ), ];

# Fall and Spring orders consistency check
# Assumption: All orders must be either Fall or Spring!
catalog_data[ , "ordhist_new" ] = catalog_data[ , "ordhist" ];

season_inconsistency = ( catalog_data[ , "ordhist" ] <
                           catalog_data[ , "falord" ] + catalog_data[ ,
"sprord" ] );

catalog_data[ season_inconsistency & catalog_data[ , "yearlp" ] < 2009,
"ordhist_new" ] = rowSums( catalog_data[ season_inconsistency &
catalog_data[ , "yearlp" ] < 2009, c( "falord", "sprord" ) ] );
```

```
# Check for order and sales history inconsistency from 2009 onwards
sums_inconsistency = catalog_data[ , "datead6" ] > "2009-01-01" & (
catalog_data[ , "ordhist" ] > rowSums( catalog_data[ , 10:13 ] ) );

catalog_data[ sums_inconsistency , "ordhist" ] = rowSums( catalog_data[
sums_inconsistency , 10:13 ] );
catalog_data[ sums_inconsistency , "slshist" ] = rowSums( catalog_data[
sums_inconsistency , 5:8 ] );

#### Univariate Outlier Removal ####

uni_outliers = outlier( catalog_data[ , c( 2:3, 5:16 ) ], logical = T
);
uni_outliers = ( rowSums( uni_outliers ) > 0 );

catalog_data = catalog_data[ !uni_outliers, ];
```

datasets creation.r

```
#### Variable Creation ####

months_ref = "2012-09-01";

# Dataset for Logistic Models

cat_logit = catalog_data[ , -c( 2:4, 17 ) ];

# Months since last purchase, and number of months between last
purchase and date created
cat_logit[ , "recency_months" ] = months_diff( catalog_data[ ,
"datelp6" ], months_ref );
cat_logit[ , "loyalty_months" ] = months_diff( catalog_data[ ,
"datead6" ], catalog_data[ , "datelp6" ] );

# Binary variables
cat_logit[ , "ordhist_bin" ] = as.numeric( cat_logit[ , "ordhist" ] > 0
);
cat_logit[ , "ordtyr_bin" ] = as.numeric( cat_logit[ , "ordtyr" ] > 0
);
cat_logit[ , "ordlyr_bin" ] = as.numeric( cat_logit[ , "ordlyr" ] > 0
);
cat_logit[ , "ord2ago_bin" ] = as.numeric( cat_logit[ , "ord2ago" ] > 0
);
cat_logit[ , "ord3ago_bin" ] = as.numeric( cat_logit[ , "ord3ago" ] > 0
);
cat_logit[ , "falord_bin" ] = as.numeric( cat_logit[ , "falord" ] > 0
);
cat_logit[ , "sprord_bin" ] = as.numeric( cat_logit[ , "sprord" ] > 0
);
cat_logit[ , "targdol_bin" ] = as.numeric( cat_logit[ , "targdol" ] > 0
);

cat_logit[ , "slsPerOrd" ] = 0;
cat_logit[ cat_logit[ , "ordhist" ], "slsPerOrd" ] = cat_logit[
cat_logit[ , "ordhist" ], "slshist" ] / cat_logit[ cat_logit[ ,
"ordhist" ], "ordhist" ];
```

```
cat_logit_test = cat_logit[ catalog_data[ , "train" ] == 0, ];
cat_logit = cat_logit[ catalog_data[ , "train" ] > 0, ];

# random_data = cat_logit[ cat_logit$targdol_bin == 0, ];
# random_data = random_data[ sample( nrow( random_data ), sum(
cat_logit$targdol_bin ) ), ];
# cat_logit2 = data.frame( rbind( cat_logit[ cat_logit$targdol_bin > 0,
], random_data ) );

# Dataset for Linear Models

cat_lm = catalog_data[ , -c( 2:4, 17 ) ];
cat_lm[ , "recency_months" ] = months_diff( catalog_data[ , "datelp6"
], months_ref );
cat_lm[ , "loyalty_months" ] = months_diff( catalog_data[ , "datead6"
], catalog_data[ , "datelp6" ] );

cat_lm[ , "slsPerOrd" ] = 0;
cat_lm[ cat_lm[ , "ordhist" ], "slsPerOrd" ] = cat_lm[ cat_lm[ ,
"ordhist" ], "slshist" ] / cat_lm[ cat_lm[ , "ordhist" ], "ordhist" ];

cat_lm_test = cat_lm[ cat_lm[ , "targdol" ] & !catalog_data[ , "train"
], ];
cat_lm = cat_lm[ cat_lm[ , "targdol" ] & catalog_data[ , "train" ], ];
```

logistic.r

```
#### Fit Logistic Models ####

# Model with sqrt order predictors ####

fitlogit_null = glm( targdol_bin ~ 1, family = "binomial", data =
cat_logit );

fitlogit_step1 = step( fitlogit_null, scope = ~ sqrt( falord ) + sqrt(
falord ):sqrt( sprord ) + sqrt( slshist ) + sqrt( ordtyr ) * sqrt(
ordlyr ) + sqrt( ord2ago ) + sqrt( ord3ago ) + sqrt( loyalty_months ) +
sqrt( recency_months ) + sqrt( slsPerOrd ), direction = "both" );

summary( fitlogit_step1 );

print_table( fitlogit_step1 );
print_table( fitlogit_step1, level = 0.227 );
print_table( fitlogit_step1, cat_logit_test, 0.227 );

# Best model with binary order predictors and log sales predictors ####

fitlogit_null = glm( targdol_bin ~ 1, family = "binomial", data =
cat_logit );

fitlogit_step2 = step( fitlogit_null, scope = ~ falord_bin +
sprord_bin:falord_bin + log( slshist + 1 ) + ordtyr_bin * ordlyr_bin +
ord2ago_bin + ord3ago_bin + sqrt( loyalty_months ) + sqrt(
recency_months ) + sqrt( slsPerOrd ), direction = "both" );
```

```
summary( fitlogit_step2 );

print_table( fitlogit_step2 );
print_table( fitlogit_step2, level = 0.1 );
print_table( fitlogit_step2, cat_logit_test, 0.1 );

#### Final Logistic Models ####

fitlogit1 = glm( targdol_bin ~ sqrt( recency_months ) + sqrt(
loyalty_months ) + sqrt( falord ):sqrt( sprord ) + sqrt( falord ) +
sqrt( ordtyr ) * sqrt( ordlyr ) + sqrt( ord2ago ) #+ sqrt( slshist )
                   , family = "binomial", data = cat_logit );

summary( fitlogit1 );
vif( fitlogit1 );

print_table( fitlogit1 );
print_table( fitlogit1, level = 0.227 );
print_table( fitlogit1, cat_logit_test, 0.226 );

1 - pchisq( fitlogit1$null.deviance - fitlogit1$deviance, df =
fitlogit1$df.null - fitlogit1$df.residual );

fitlogit2 = glm( targdol_bin ~ sqrt( recency_months ) + sqrt(
loyalty_months ) + ordtyr_bin + falord_bin + ord3ago_bin + ord2ago_bin
+ log( slshist + 1 ) + ordlyr_bin + ordtyr_bin:ordlyr_bin, family =
"binomial", data = cat_logit );

summary( fitlogit2 );
vif( fitlogit2 );

print_table( fitlogit2 );
print_table( fitlogit2, level = 0.1 );
print_table( fitlogit2, cat_logit_test, 0.1 );

1 - pchisq( fitlogit2$null.deviance - fitlogit1$deviance, df =
fitlogit2$df.null - fitlogit2$df.residual );

plot.roc( cat_logit[ , "targdol_bin" ], fitted( fitlogit_null ) );
plot.roc( cat_logit[ , "targdol_bin" ], fitted( fitlogit1 ) , add = T,
col = 2 );
plot.roc( cat_logit[ , "targdol_bin" ], fitted( fitlogit2 ) , add = T,
col = 3 );
```

linear models.r

```
# Fit Linear Models

fit_null = lm( log( targdol + 1 ) ~ 1, data = cat_lm );

#### Model with sqrt predictors ####

fit_step1 = step( fit_null, scope = ~ sqrt( ordhist ) + sqrt( slshist )
+ sqrt( ordtyr * ordlyr ) + sqrt(   ordtyr ) + sqrt( ordlyr ) + sqrt(
slstyr * slslyr ) + sqrt( slstyr ) + sqrt( slslyr ) + recency_months +
sqrt( slsPerOrd ), direction = "both" );
```

```
summary( fit_step1 );
vif( fit_step1 );

MSEP( fitlogit1, fit_step1, cat_logit_test, 0.227 );
fin_criterion( fitlogit1, fit_step1, cat_logit_test, level = 0.227 );

fit_reg1 = regsubsets( log( targdol + 1 ) ~ sqrt( ordhist ) + sqrt(
slshist ) + sqrt( ordtyr * ordlyr ) + sqrt( ordtyr ) + sqrt( ordlyr ) +
sqrt( slstyr * slslyr ) + sqrt( slstyr ) + sqrt( slslyr ) +
recency_months + sqrt( slsPerOrd ), data = cat_lm, method =
"exhaustive" );

plot( fit_reg1 );

fit1 = lm( log( targdol + 1 ) ~ sqrt( slshist ) + sqrt( ordhist ) +
sqrt( slstyr * slslyr ), data = cat_lm );

summary( fit1 );
vif( fit1 );

MSEP( fitlogit1, fit1, cat_logit_test, 0.2278 );
fin_criterion( fitlogit1, fit1, cat_logit_test, 0.2278 );

avPlots( fit1 );
crPlots( fit1 );

n = nrow( cat_lm );
p = fit1$rank - 1;

# Infulential + Outlier Removal using Cook's Distance
cutoff = 4 / ( n - p - 1 );
outliers = which( cooks.distance( fit1 ) > cutoff );

influencePlot( fit1 );
plot( fit1, which = 4, cook.levels = cutoff );

cat_lm = cat_lm[ -outliers, ];

fit1 = lm( log( targdol + 1 ) ~ sqrt( slshist ) + sqrt( ordhist ) +
sqrt( slstyr * slslyr ), data = cat_lm );

summary( fit1 );
vif( fit1 );

MSEP( fitlogit1, fit1, cat_logit_test, 0.2278 );
fin_criterion( fitlogit1, fit1, cat_logit_test, 0.2278 );

#### Model with inverse transforms on orders and logarithmic transforms
on sales predictors ####

fit_null = lm( log( targdol + 1 ) ~ 1, data = cat_lm );

fit_step2 = step( fit_null, scope = ~ I( 1 / ( ordhist + 1 ) ) + log(
slshist + 1 ) + I( 1 / ( ord2ago + 1 ) ) + loyalty_months  + log(
slstyr * slslyr + 1 ) + log( slstyr + 1 ) + log( slslyr + 1 ) + log(
slsPerOrd + 1 ), direction = "both" );
```

```
summary( fit_step2 );
vif( fit_step2 );

MSEP( fitlogit2, fit_step2, cat_logit_test, 0.227 );
fin_criterion( fitlogit2, fit_step2, cat_logit_test, 0.227 );

crPlots( fit_step2 );

fit_reg2 = regsubsets( log(targdol + 1) ~ I( 1/( ordhist + 1 ) ) + log(
slshist + 1 ) + I( 1/( ord2ago + 1 ) ) + loyalty_months  + log( slstyr
* slslyr + 1 ) + log( slstyr + 1 ) + log( slslyr + 1 ) + log( slsPerOrd
+ 1 ), data = cat_lm, method = "exhaustive" );

plot( fit_reg2 );

fit2 = lm( log( targdol + 1 ) ~ I( 1 / ( ordhist + 1 ) ) + log( slshist
+ 1 ) + log( slstyr * slslyr + 1 ), cat_lm );

summary( fit2 );
vif( fit2 );

MSEP( fitlogit2, fit2, cat_logit_test, 0.2297 );
fin_criterion( fitlogit2, fit2, cat_logit_test, 0.2297 );

avPlots( fit2 );
crPlots( fit2 );

n = nrow( cat_lm );
p = fit2$rank - 1;

# Infulential + Outlier Removal using Cook's Distance
cutoff = 4 / ( n - p - 1 );
outliers = which( cooks.distance( fit2 ) > cutoff );

influencePlot( fit2 );
plot( fit2, which = 4, cook.levels = cutoff );

cat_lm = cat_lm[ -outliers, ];

fit2 = lm( log( targdol + 1 ) ~ I( 1 / ( ordhist + 1 ) ) + log( slshist
+ 1 ) + log( slstyr * slslyr + 1 ), cat_lm );

summary( fit2 );
vif( fit2 );

MSEP( fitlogit2, fit2, cat_logit_test, 0.2297 );
fin_criterion( fitlogit2, fit2, cat_logit_test, 0.2297 );
```